UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/779,503	02/13/2004	David Arthur James Webb JR.	200304950-1	9855

7590 03/28/2005

Hewlett-Packard Company
Intellectual Property Administration
P.O. Box 272400
Fort Collins, CO 80527-2400

EXAMINER	
HARKNESS, CHARLES A	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 03/28/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/779,503	JAMES WEBB ET AL.	
	Examiner	Art Unit	
	Charles A Harkness	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *13 February 2004*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-19* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-19* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *13 February 2004* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *2/13/04*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

TC

DETAILED ACTION

Specification

1. The applicant or their representatives are urged to review the specification and submit corrections for all mistakes of a grammatical, clerical, or typographical nature.

Double Patenting

The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. See *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970);and, *In re Thorington,* 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent is shown to be commonly owned with this application. See 37 CFR 1.130(b).

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

2. Claims 1-19 are rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-19, respectively, of U.S. Patent No. 6,738,896. Although the conflicting claims are not identical, they are not patentably distinct from each other because although the instant application does not claim that the issuing of a program step is "random" in nature, does not mean that the instant application is patentably distinct. Simply removing features from an invention, or not mentioning them does not constitute a patentable invention. As shown in In re Karlson, **153 USPQ 184 (CCPA 1963)** elimination of an element or its function generally is not given patentable weight or would have been obvious improvements. Also, it would have been obvious to one of ordinary skill in the art at the time of the invention that to perform a calculation to determine the status bit of a location in a queue that allows to

"wrap-around" a modulus calculation is involved, and that requires a mod base of the total number of entries in the queue. Therefore, to determine the status bit of the location in the queue, you need to know the total number of locations in the queue. Thus, it would have been obvious to one of ordinary skill in the art at the time of the invention to use the value of the total number of locations in the queue in the calculation of the status, or color bit.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

3. Claims 1-19 are rejected under 35 U.S.C. 103(a) as being unpatentable over Akkary et al, U.S. Patent Number 6,182,210 (herein referred to as Akkary) in view of Popescu et al. (IEEE Micro, Vol. 11 Issue 3, Jun 1991).

4. Referring to claims 1 and 10 Akkary has taught a method for executing program instructions comprising the steps of:

assigning each individual ones of a plurality of program steps a unique number (Akkary figure 2 and 24 column 9 lines 6-10, column 16 lines 42-49, column 18 lines 57-58; the reason for assigning IDs to the instruction is for maintaining a record of the original program order so you can piece things back together properly later. Also, if the usage of the ID's is to determine age and/or to piece things back together after out of order execution, then they would need to be

in an order and unique, otherwise you couldn't determine the original order unless there was some pattern to the assignment; the same reasoning is used for assigning load Ids);

issuing a program step to an execution queue (Akkary figure 2 and 24 column 16 lines 42-49; some would go from the sched/issue unit 156 to the exec units 158 and some to the load or store buffers, 182 and 184; based on Applicant's definition of random, see specification page 9 lines 6-11);

selecting a specific one of a plurality of numbered locations in the execution queue based upon the unique number of the issued program step, each numbered location having an instruction valid bit and the execution queue having a certain total number of numbered locations (Akkary figure 2 and 24 column 18 line 54-column 19 line 23, column 24 lines 25-34; the load queue would have to have a pointer or counter, or both to keep track of which spot is the next location in the queue that the next sequential load instruction will be stored, therefore the next sequential instruction ID, or load buffer ID, will be selected and then selecting the specific location in the queue that is the next location that will be available when a load instruction is retired in program order);

determining the value of the instruction valid bit associated with the selected numbered location (Akkary figure 2 and 24 column 18 line 54-column 19 line 23, column 24 lines 25-34; inherently the load queue would have to check the valid bit before it stored the next load instruction in the location, otherwise it would overwrite the current load instruction when it is still valid);

determining availability of the selected numbered location such that the issued program step is stored in the selected numbered location if the selected numbered location is available

(Akkary figure 2 and 24 column 18 lines 58-62), and a full flag for the execution queue is issued if the selected numbered location is not available (Akkary figure 2 and 24 column 19 lines 14-16 and lines 31-37; the load queue must send some indicator to the rename/allocate unit 150 if it cannot accept anymore load instructions, otherwise the system would not work properly since the rename/allocate unit 150 would continue to send instructions and would overwrite instructions that have not yet been retired; this would be based on the comparing the valid entry bits).

Akkary has not taught wherein:

using the certain total number and the unique number of the issued program step, calculating value of a status bit for the selected numbered location; and

based upon the determined value of the instruction value bit and the calculated value of the status bit, determining availability of the selected numbered location.

5. However, Akkary has taught using a valid entry field to determine if the location in the queue is a valid instruction, and if the queue is full not allowing more instructions to be allocated to the queue (Akkary figure 2 and 24 column 19 lines 14-16 and lines 31-37). Akkary has also taught using a head pointer and tail pointer in the queue (Akkary figure 2 and 24 column 24 lines 25-29).

Popescu et al. has taught wherein calculating value of a status bit, or a color bit, for the selected numbered location, using the certain total number and the unique number of the issued program step and based upon the determined value of the instruction value bit and the calculated value of the status bit, determining availability of the selected numbered location. Popescu et al. taught that the index position of the instruction in a queue wraps around from its maximum value back to its minimum value, and using a 'color' bit to determine where the older instructions start by

flipping the 'color' bit when the index of the queue wraps around (Popescu et al. page 64 paragraph 4; In order for an index to wrap around, there must inherently be a modulus operation of the location number occurring, otherwise if you have a 16 entry queue, and you get to the 17^{th} instruction, and the last entry is filled, you would have to wait for the last entry to become available; in this manner, allowing the queue to wrap around to the top, instructions can continually be sent to the load queue as the oldest instructions, or locations, retire and become available – also, the total number of queue entries would have to be includes since the modulus calculation would be based on the total number of locations available). A numeric calculation would have to be done to see which location the next instruction will be selected to go. This calculation would include a modulus type function since the queue wraps around to the beginning once it reaches the end. The color bit would be the status bit, and the valid entry field of Akkary would still need to be checked to see if that specific location in the queue, or buffer, is available. It would have been obvious to one of ordinary skill in the art at the time of the invention to use a wrap-around queue as taught by Popescu et al. to allow instructions to continually be issued to the load queue as resources and/or locations become available. This will increase the throughput of the load queue, and thus the throughput for all the instructions of the program since some instructions will be dependent on the load instructions. Having a queue that wraps around will also prevent the queue from 'shifting' the entries in the location up to the beginning every time an instruction is retired. This will reduce the amount of hardware required for the unit. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to select a location by calculating and recording a modulus of the location number to increase throughput of the system and to reduce the hardware needed in the system.

6. Referring to claims 2 and 12 Akkary has not taught the method of claim 1, wherein the unique number is a monotonically ascending series of integers assigned to the program steps in the same order as the program is executed (Akkary column 9 lines 6-10, column 16 lines 42-49, column 18 lines 57-58). However, the use of a monotonically increasing integer to be a unique number for each program step would have been obvious. A strong motivation at the time of invention for using a monotonically increasing integer is to simplify the logic necessary to create the unique number. One of ordinary skill in the art at the time of invention would have known that one could simply use an integer counter to generate the unique ID at the time the instructions are issued and would have been motivated to do so to reduce chip cost.

7. Referring to claims 3 and 13 Akkary has taught the method of claim 1, wherein the execution queue is one of a load queue and a store queue (Akkary figure 2 reference numbers 182 and 184).

8. Referring to claims 4 and 14 has taught the method of claim 1, wherein the program step is issued to the execution queue in an order determined by an availability of a selected one of a plurality of computation resources (Akkary figure 2 column 18 line 54-column 19 line 8, column 24 lines 25-34; the load instructions are issued to the load queue when resources and locations are available, after a load has retired and been deallocated using the valid and/or status bits).

9. Referring to claims 5 and 15 Akkary has taught the method of claim 1, wherein each program step issued to the execution queue is assigned a queue location number based upon ascending value of the unique number assigned to the program step (Akkary column 9 lines 6-10, column 16 lines 42-49, column 18 lines 57-58). It is not clear what the ordered list of the

program steps refers to since there is no prior reference to any ordered lists in the claims. For the purpose of this examination, it will be assumed "ordered list" refers to a queue.

However, the use of a monotonically increasing integer to be a unique number for each program step would have been obvious. A strong motivation at the time of invention for using a monotonically increasing integer is to simplify the logic necessary to create the unique number. One of ordinary skill in the art at the time of invention would have known that one could simply use an integer counter to generate the unique ID at the time the instructions are issued and would have been motivated to do so to reduce chip cost.

10. Referring to claims 6 and 16 Akkary and Popescu has taught the method of claim 5, wherein the assigned queue location number is one of a plurality of numbered locations in each one of a plurality of execution queues and is selected by calculating and recording a modulus of the queue location number. Refer to the rejection of claims 1 and 10. Popescu has taught wherein the specific one of the plurality of numbered locations in each one of the plurality of execution queues is selected by calculating and recording a modulus of the location number. Popescu et al. taught that the index position of the instruction in a queue wraps around from its maximum value back to its minimum value (Popescu et al. page 64 paragraph 4; In order for an index to wrap around, there must inherently be a modulus operation of the location number occurring, otherwise if you have a 16 entry queue, and you get to the 17^{th} instruction, and the last entry is filled, you would have to wait for the last entry to become available; in this manner, allowing the queue to wrap around to the top, instructions can continually be sent to the load queue as the oldest instructions, or locations, retire and become available).

11. Referring to claims 7 and 17 Akkary has taught the method of claim 6, wherein the divisor of the modulus calculation is equal to a number of locations in the queue (Popescu et al. page 64 paragraph 4; Inherently for the modulus operation to allow the location numbers to wrap around the queue, the divisor of the modulus must be the number of locations, otherwise, when the modulus is used to decide which location to put an instruction it may not calculate a location that exists in the queue).

12. Referring to claims 8 and 18 Akkary has taught the method of claim 7, wherein the status bit of the selected numbered location is switched when said location becomes invalid (Akkary column 19 lines 14-16; the bit would have to be switched from valid to invalid when it becomes available, otherwise no new load instructions could be issued to the queue because all of the locations would be marked valid).

13. Referring to claims 9 and 19 Akkary has taught the method of claim 1, wherein a load queue is part of the execution queue (Akkary figure 2 reference numbers 182 and 184). For the purpose of this examination, it is assumed that Applicant meant "the load queue and store queue are included in the execution queue).

14. Referring to claim 11 Akkary has taught the apparatus of claim 10, further comprising a plurality of computer systems, all with access to a plurality of computation resources (For the purpose of this examination, it is assumed that Applicant is stating that they will be implementing their invention in many computer systems. Official Notice is taken that it would have been obvious to one of ordinary skill in the art at the time of the invention to implement an invention into more than one computer system so that Applicant can take the most advantage of their invention in economical matters).

Conclusion

The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

U.S. Patent Number 5, 751, 985 Shen et al., has taught a processor structure and method for tracking instruction status to maintain precise state.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Charles A Harkness whose telephone number is 571-272-4167. The examiner can normally be reached on 9Flex.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on 571-272-4162. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Charles Allen Harkness

Examiner

Art Unit 2183

March 3, 2005

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100